Cost-Bandwidth Tradeoff In Distributed Storage Systems

Soroush Akhlaghi, Abbas Kiani and Mohammad Reza Ghanavati
Department of Electrical Engineering
Shahed University
Tehran, Iran
Email: {akhlaghi,akiani,ghanavati}@shahed.ac.ir

***Abstract*—Distributed storage systems are mainly justified due to the limited amount of storage capacity and improving the reliability through distributing data over multiple storage nodes. On the other hand, it may happen the data is stored in unreliable nodes, while it is desired the end user to have a reliable access to the stored data. So, in an event that a node is damaged, to prevent the system reliability to regress, it is necessary to regenerate a new node with the same amount of stored data as the damaged node to retain the number of storage nodes, thereby having the previous reliability.**

This requires the new node to connect to some of existing nodes and downloads the required information, thereby occupying some bandwidth, called the repair bandwidth. On the other hand, it is more likely the cost of downloading varies across different nodes. This paper aims at investigating the theoretical cost-bandwidth tradeoff, and more importantly, it is demonstrated that any point on this curve can be achieved through the use of the so called generalized regenerating codes which is an enhancement of the regeneration codes introduced by Dimakis et al. in [1].

I. Introduction

Data in distributed storage systems should be stored reliably for a long period of time. This is due to the need for surviving in the case that individual failures occur, thus having a long-term durability. To this end, the system should have the possibility of self-repairing in the case that a node is failed or leaves the system. This requires a great deal of data transferring due to repairing a failure node, called repair bandwidth. In some cases, a great deal of repair bandwidth is consumed to construct a new node.

To have a reliable data, various strategies have been proposed which basically attempt to add some redundancy bits to the original data and distributing the encoded data across distinct nodes in an effective manner. The simplest strategy is replication in which each node stores the original data file, hence, the data of one node is adequate to reconstruct the original data. However, this is not a wise method due to the need to a high storage capacity. To address this issue, in [2], [3] instead of exploiting naive replication code, an erasure coding is used in which the original data file of size M is divided into k pieces of size M/k, and encoded into n data fragments to be stored in one of existing n nodes. The encoding process is such that having access to the stored data of k nodes is adequate to reconstruct the original data. In other words, a new node should be connected to k nodes to have an access to all information. As a result, for a large value of k, the storage capacity of each node is dramatically reduced as compared to the replication code, since instead of storing data size of M, we need to merely store a fragment of data size M/k at each node [4], [5]. Although, the erasure code requires the same repair bandwidth as compared to the replication code and imposes a decoding complexity into the system, it makes a balance between the system reliability and redundancy.

To take the advantages of both replication (simple decoding method) and erasure coding (low storage capacity), in [4] a hybrid strategy is proposed. This strategy uses a single node containing an exact replica of the original data file as well as some nodes with the structure of erasure coding. Thus, for generating a new data fragment, this replica is used and just a data of size M/k is transferred across the network. Although the repair bandwidth of the hybrid strategy is reduced, the system complexity is greatly increased, i.e., if the replica is failed, creating a new fragment is deferred until restoring the replica. This in turn, may not be feasible when there is a stringent delay constraint.

This motivated Dimakis et al. in [1] to deduce an elegant coding strategy, dubbed regenerating codes (RC), to reduce the repair bandwidth without the use of replica. It is shown that for creating a new data fragment, the newcomer node should be connected to d nodes ($d \geq k$) and download β bits from each surviving nodes. Accordingly, a trade-off between storage per node and repair bandwidth ($d\beta$) is identified.

Regenerating codes and other existing methods are motivated by the assumption that surviving nodes have equal download cost, and creating a new node is accomplished through downloading the same amount of information from each surviving node. However, it may happen there is a different cost associated with each node. Thus, in an attempt to replace a damaged node with a new node, one may want to make a balance between the download cost and the repair bandwidth.

The current study aims to address the aforementioned issue when there are two sets of nodes, each having different download costs, while the nodes of each set have the same cost. However, the material in this paper can be readily extended to more general cases. Accordingly, it is assumed a newcomer node downloads β_1 and β_2 bits, respectively, from each surviving node of cost C_1 and C_2, where it is simply assumed $C_1 \leq C_2$. It will be later shown that under certain conditions, if β_1 is larger than β_2, the total download cost is

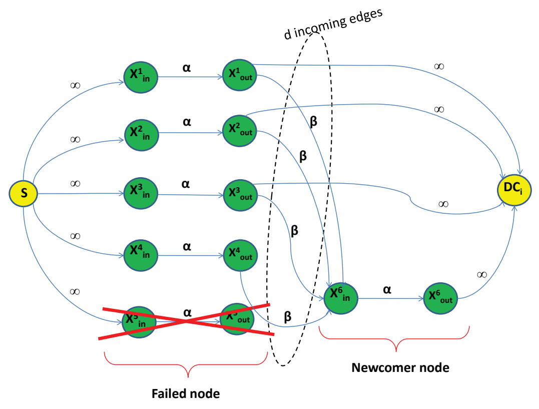

Fig. 1. An example of *Information Flow Graph* when a failure is occurred (it is marked by cross lines), thus a new node is initiated.

reduced at the expense of increasing the repair bandwidth. In other words, the more β_1 is larger than β_2, the less download cost is produced, while having more repair bandwidth as if $\beta_1 = \beta_2$. Moreover, for a given β_1 and β_2, congruent to what is done in [1], a trade-off between the storage per node and repair-bandwidth is identified.

The rest of paper is organized as follows: In section II, distributed storage systems are briefly introduced and their equivalent *Information Flow Graph* is introduced. Accordingly, it is argued that network coding can approach the capacity of such systems. Finally, regenerating codes are motivated and briefly introduced. Section III states the problem formulation and motivates the main idea, finally gives an overview of the approach. Sections VI,VII, present numerical results and conclude the paper, respectively.

II. Background

A. Distributed Storage Systems and connection to Network Coding

In distributed storage systems, nodes join or leave the network continuously, hence, the network configuration varies across time. Motivated by the pioneering work in [1], this network can be thought as an *information flow graph*, a directed acyclic graph consisting of three types of nodes: (i) A single source node (S), (ii) Some intermediate nodes and (iii) Data collectors (DC nodes). The source node is the source of original data file, intermediate nodes are storage nodes and each data collector corresponds to a request for reconstructing original data file. Each storage node is represented by pairs of incoming and outgoing nodes connected by a directional edge whose capacity is the corresponding storage capacity of this storage node. In this work, we simply assume all storage nodes are of capacity α. Moreover, it is assumed edges departing the storage nodes and arriving to a DC node have infinite capacity. This reflects the fact that DC nodes have access to all stored data of the surviving nodes they are connected to.

As is mentioned earlier, the corresponding *information flow graph* evolves constantly across time to reflect any changes happening throughout the network. This graph starts from the source node, indicating it is the only active node at the first step. Then, assuming the total number of storage nodes is n, the source node divides the original data file of size M into k pieces, encodes these k pieces to n data fragments each to be stored in one of existing storage nodes through direct edges of infinite capacity. In the case that a storage node leaves the system or a failure occurs, this node is replaced by a new one, called the newcomer node. The newcomer connects to d active nodes out of $n-1$ existing nodes and downloads β bits from each. Accordingly, the corresponding *information flow graph* is updated through establishing d directed edges of capacity β, starting from outgoing nodes affiliated to the selected storage nodes and terminating to the corresponding incoming node of the newcomer (Fig.1). In this case, the total information received by the newcomer node, $d\beta$, is called the repair bandwidth (γ). Finally, the data is reconstructed at each DC node through connecting to any arbitrary set of k nodes (storage nodes), including the newcommer nodes. The edges connecting the selected storage nodes to the corresponding DC node are assumed to be of infinite capacity.

Incorporating the graphical representation of distributed storage systems gives the opportunity to relate the storage capacity as well as repair bandwidth of the original problem to some characteristics of the corresponding *information flow graph*. Specifically, we are interested in an important quantity, call the network throughput introduced by Ahlswede et al. in [6], which basically identifies the maximum allowable information flow from a source to a destination node, assuming each link is subject to a limited capacity. Accordingly, it is demonstrated that using a proper coding at intermediate nodes, it is possible to get the information with a throughput at most equal to what is promised by the so called min-cut theorem [6]. This is achieved through using an elegant coding strategy, called network coding, which basically can approach the multicast capacity of such networks [7], [8]. The notion of using network coding has beaten the previous belief of using simple routing mechanism at intermediate nodes.

B. Regenerating Codes

As is mentioned earlier, for erasure coding, having an access to the data of k storage nodes out of existing n nodes is adequate to reconstruct the original data file. Thus, the newcomer needs to connect to exactly $d = k$ nodes and downloads all of stored data ($\alpha = M/k$), thus $\beta = \alpha = M/k$. So the repair bandwidth becomes the same as the size of data file, i.e., $\gamma = d\beta = M$. On the other hand, Dimakis et al. in [1] show that if a newcomer could connect to more than k surviving nodes and downloads a certain function of their stored information, a lower repair bandwidth would be achieved, while having the same storage capacity as compared to that of erasure coding.

To this end, it is shown the task of computing the repair bandwidth can be translated to a multicast problem over the corresponding information flow graph for which an optimal trade-off between the storage per node, α, and the repair bandwidth, γ, is identified. This optimal trade-off curve includes two extremal points corresponding to the minimum storage capacity per node and minimum repair bandwidth, respectively. Recall that any points on the trade-off curve, including the extremal points can be achieved by the use of network coding approach. The former, minimum storage capacity, is achieved by use of the so called Minimum Storage Regenerating (MSR) codes. The latter, is realized through using Minimum Bandwidth Regenerating (MBR) codes. Accordingly, the corresponding storage capacity per node (α) and repair bandwidth (γ) for MSR and MBR codes are computed as follows [1]:

$$\begin{aligned}(\alpha_{MSR}, \gamma_{MSR}) &= (\frac{M}{k}, \frac{Md}{k(d-k+1)}) \\ (\alpha_{MBR}, \gamma_{MBR}) &= (\frac{2Md}{2kd-k^2+k}, \frac{2Md}{2kd-k^2+k})\ , \quad (1)\end{aligned}$$

where in (1), it is assumed the total data file is of size M. Moreover, d denotes the number of storage nodes which a newcomer is connected to ($d \geq k$), and k represents the total number of nodes which are required to reconstruct the original data file. In other words, a DC node needs to connect to exactly k storage nodes to reconstruct the original data file.

III. Problem formulation and the proposed method

MSR and MBR codes are motivated by the assumption that the download cost of all storage nodes are the same. However, we rely on a more realistic situation in which storage nodes are subject to different download costs and the download cost is of great concern. Specifically, we concentrate on the case that there are totally two sets of storage nodes S_1 and S_2 with download costs per information bit equal to C_1 and C_2, respectively[1]. Accordingly, in regenerating codes, a newcomer connects to d nodes, each belongs either to S_1 or S_2. Assuming d_1 nodes are of cost C_1 and $d_2 = d - d_1$ nodes are of cost C_2, thus the total cost for reconstructing a damaged node becomes:

$$C_T = (C_1 d_1 + C_2 d_2)\beta\ , \tag{2}$$

where β is the total information downloaded from each node. Equation (2) indicates that the same amount of information is downloaded from each node, no matter which set it basically belongs to. However, an important enquiry may arise; How to make a balance between the repair bandwidth and the total cost?. In this work, we aim at addressing the aforementioned issue and more importantly, to establish a trade-off between the repair bandwidth, the storage capacity, and the total cost.

We employ a variation of the regenerating code, dubbed Generalized Regenerating Code (GRC), in which the newcomer downloads different amount of information depending on the type of storage node. In the course of downloading, we consider there are totaly d_1 nodes with download cost C_1 and d_2 nodes ($d_2 = d - d_1$) with download cost C_2 ($C_2 \geq C_1$), where β_1 and β_2 bits are downloaded from each of these nodes, respectively. Noting $C_2 \geq C_1$, one can get a lower cost if $\beta_1 \geq \beta_2$. Throughout the paper, we assume $\beta_1 = k'\beta_2$ [2].

[1]This enables the problem can be mathematically tractable. However, one can readily follow the same approach for more general cases.

[2]It is worth mentioning that for some practical purposes, k' should take an integer value.

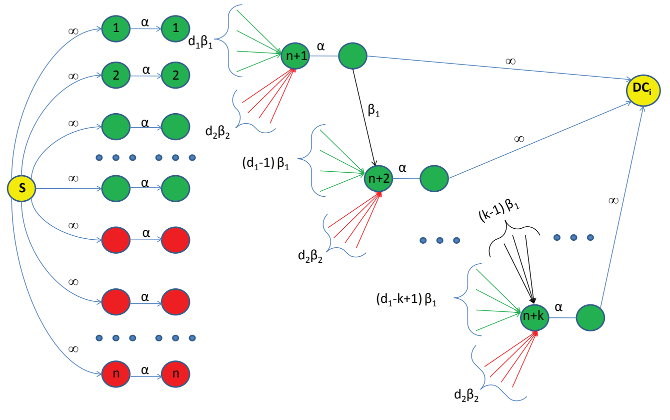

Fig. 2. $\mathcal{G}^*$ for $d_1 \geq k$

As a result, the total cost for constructing a new node in this strategy is as follows:

$$C_T = C_1 d_1 \beta_1 + C_2 d_2 \beta_2\ . \tag{3}$$

It should be noted that, as is shown in the next sections, k' is inversely proportional to the relative download cost, meaning the larger k' results in the less relative cost of GRC as compared to that of the regenerating codes. Then, for a given k', the problem is translated to computing β_2 (or equivalently β_1) for which the minimum repair bandwidth or minimum storage capacity per node is obtained. Accordingly, It is shown even more reduction in C_T is possible at the expense of increasing the repair bandwidth. In the next section, we examine two different scenarios of $d_1 \geq k$ and $d_1 < k$ to explore the problem.

IV. Scenario A: $d_1 \geq k$

Consider any given finite information flow graph $\mathcal{G}$, with a finite set of data collectors. In [1], it is argued that "If the minimum of the min-cuts separating the source with each data collector is larger or equal to the data object size M, then there exists a linear network code defined over a sufficiently large finite field F (whose size depends on the graph size) such that all data collectors can recover the data object". In Fig.2, the graph $\mathcal{G}^*$, a portion of the corresponding *Information flow graph* $\mathcal{G}$, entailing the minimum of the min-cuts for $d_1 \geq k$ is shown. So referring to this flow graph and noting the above argument, the following condition is necessary to reconstruct the original data file:

$$\sum_{i=0}^{k-1} \min\{(d_1\beta_1 + d_2\beta_2 - i\beta_1),\ \alpha\} \geq M\ . \tag{4}$$

Thus, using (4) and noting $\beta_1 = k'\beta_2$, and after some manipulations, a tradeoff between α_{min} (the minimum required storage) and β_2 is identified as follows,

$$\alpha_{min}(d_1, d_2, k', \beta_2) = \begin{cases} \frac{M}{k} & \beta_2 \in \big[f(0), \infty\big) \\ \frac{2M - g(i)\beta_2}{2(k-i)} & \beta_2 \in \big[f(i), f(i-1)\big)\ , \end{cases} \tag{5}$$

where

$$\begin{aligned} f(i) &\triangleq \frac{2M}{2k(d_1k' + d_2 - kk') + k'(i+1)(2k-i)} \\ g(i) &\triangleq i(2d_1k' + 2d_2 - 2kk' + (i+1)k') \ . \end{aligned} \quad (6)$$

Thus, $\beta_{2_{min}}$ (the minimum required download from each node) can be computed as,

$$\begin{aligned} \beta_{2_{min}} &= f(k-1) \\ &= \frac{2M}{k(2d_1k' + 2d_2 - kk' + k')} \ . \end{aligned} \quad (7)$$

In other words, for any $\alpha \geq \alpha_{min}(d_1, d_2, k', \beta_2)$, the points $(n, k, d_1, d_2, \alpha, \beta_1, \beta_2)$ with linear network coding are achievable.

Thus, the tradeoff curve between the storage capacity (α) and the repair bandwidth ($\gamma = \beta_1 d_1 + \beta_2 d_2$) can be established through using (5), where $\beta_1 = k'\beta_2$. This curve has two extremal points. One corresponds to minimum storage capacity and the other related to the minimum repair bandwidth. We call the codes that achieve these points as Generalized Minimum Storage Regenerating (GMSR) and Generalized Minimum Bandwidth Regenerating (GMBR) codes, respectively. GMSR is identified with the following storage capacity-repair bandwidth pair,

$$(\alpha_{\text{GMSR}}, \gamma_{\text{GMSR}}) = \big(\frac{M}{k}, \frac{M(d_2 + k'd_1)}{k(d_1k' + d_2 - kk' + k')}\big) \ . \quad (8)$$

Similarly, for GMBR, we arrive at the following,

$$(\alpha_{\text{GMBR}}, \gamma_{\text{GMBR}}) = \\ \big(\tfrac{2M(d_2+k'd_1)}{k(2d_1k'+2d_2-kk'+k')}, \tfrac{2M(d_2+k'd_1)}{k(2d_1k'+2d_2-kk'+k')}\big) \ . \quad (9)$$

It can be verified that for the special case of $k' = 1$ and $d = d_1 + d_2$, equations (8) and (9) become similar to the resulting storage capacity-repair bandwidth pairs of MSR and MBR codes [1], respectively. Also, for the case of $k' \to \infty$, noting $\beta_2 = \beta_1/k'$, one can conclude that $\beta_2 = 0$, hence, the nodes with lower download cost are merely exploited throughout the course of downloading. Accordingly, $\big(\frac{M}{k}, \frac{Md_1}{k(d_1-k+1)}\big)$ and $\big(\frac{2Md_1}{k(2d_1-k+1)}, \frac{2Md_1}{k(2d_1-k+1)}\big)$ are the corresponding storage capacity-repair bandwidth pairs of the resulting GMSR and GMBR codes. As is expected, referring to (1), these pairs are similar to that of MSR and MBR codes with $d = d_1$.

A. Comparison between GMSR and MSR when $d_1 \geq k$

Referring to (8) and the resulting storage capacity-repair bandwidth of MSR as is given in (1), GMSR and MSR yield the same storage capacity per node. However, they exhibit different repair bandwidth. To have a basis of comparison for the resulting repair bandwidth of GMSR and MSR, we define the bandwidth ratio $\rho_{\text{MSR}}(k')$ as follows,

$$\rho_{\text{MSR}}(k') \triangleq \frac{\gamma_{\text{GMSR}}(k')}{\gamma_{\text{MSR}}} = \frac{(d_2 + k'd_1)(d-k+1)}{d(d_1k' + d_2 - kk' + k')} \ . \quad (10)$$

It can be verified that as long as $d \geq k$ and $k \geq 1$, the derivation of (10) with respect to k' is positive. As these conditions hold here, $\rho_{\text{MSR}}(k')$ is an increasing function with respect to k' and more importantly, noting $\rho_{\text{MSR}}(1) = 1$, thus $\rho_{\text{MSR}}(k')$ is greater than one for $k' \geq 1$. Thus, the repair bandwidth of GMSR is greater than that of MSR. Moreover, we define the download cost ratio $\eta_{\text{MSR}}(k')$ to compare the download cost of GMSR to that of MSR, as follows,

$$\begin{aligned} \eta_{\text{MSR}}(k') &\triangleq \frac{C_{T_{\text{GMSR}}}(k')}{C_{T_{\text{MSR}}}} \\ &= \frac{(C_1d_1k' + C_2d_2)(d-k+1)}{(d_1k' + d_2 - kk' + k')(C_1d_1 + C_2d_2)} \ . \end{aligned} \quad (11)$$

Note that $\eta_{\text{MSR}}(1) = 1$. In order to have $C_{T_{\text{GMSR}}}$ lower than $C_{T_{\text{MSR}}}$, $\eta_{\text{MSR}}(k')$ should be a decreasing function, meaning to have a negative derivation with respect to k'. As a result, taking derivation of (11), one can verify that the following condition should be satisfied,

$$\frac{C_2}{C_1} \geq \frac{d_1}{d_1 - k + 1} \ . \quad (12)$$

It is worth mentioning that if the above condition holds, the minimum value of η_{MSR} is achieved as k' tends to infinity, i.e., $\eta_{\text{MSR}}(+\infty) = \frac{C_1d_1(d-k+1)}{(d_1-k+1)(C_1d_1+C_2d_2)}$.

B. Comparison between GMBR and MBR when $d_1 \geq k$

Equations (1) and (9) indicate that the storage per node is equal to the repair bandwidth for both MBR and GMBR codes. As a result, any findings for the corresponding repair bandwidths of MBR and GMBR codes, can also be considered for storage per node as well. In this regard, we define the repair bandwidth ratio $\rho_{\text{MBR}}(k')$ as follows,

$$\rho_{\text{MBR}}(k') \triangleq \frac{\gamma_{\text{GMBR}}(k')}{\gamma_{\text{MBR}}} = \frac{(d_2 + k'd_1)(2d-k+1)}{d(2d_1k' + 2d_2 - kk' + k')} \ . \quad (13)$$

Obviously, we have $\rho_{\text{MBR}}(1) = 1$. Again, following the same approach as is done in IV-A, one can readily verify that if the conditions $k \geq 1$ and $k' \geq 1$ hold, $\rho_{\text{MBR}}(k')$ is always greater than one. Thus, the repair bandwidth of GMBR is greater than that of MBR. Accordingly, we define the download cost ratio as follows,

$$\begin{aligned} \eta_{\text{MBR}}(k') &\triangleq \frac{C_{T_{\text{GMBR}}}(k')}{C_{T_{\text{MBR}}}} \\ &= \frac{(C_1d_1k' + C_2d_2)(2d-k+1)}{(2d_1k' + 2d_2 - kk' + k')(C_1d_1 + C_2d_2)} \ . \end{aligned} \quad (14)$$

Note that $\eta_{\text{MBR}}(1) = 1$. Taking derivation of $\eta_{\text{MBR}}(k')$ with respect to k', one can verify that to have $\eta_{\text{MBR}}(1) \leq 1$, the following condition should be satisfied,

$$\frac{C_2}{C_1} \geq \frac{2d_1}{2d_1 - k + 1} \ . \quad (15)$$

It is worth mentioning that the minimum value of η_{MBR} is achieved as k' tends to infinity, i.e., $\eta_{\text{MBR}}(+\infty) = \frac{C_1d_1(2d-k+1)}{(2d_1-k+1)(C_1d_1+C_2d_2)}$.

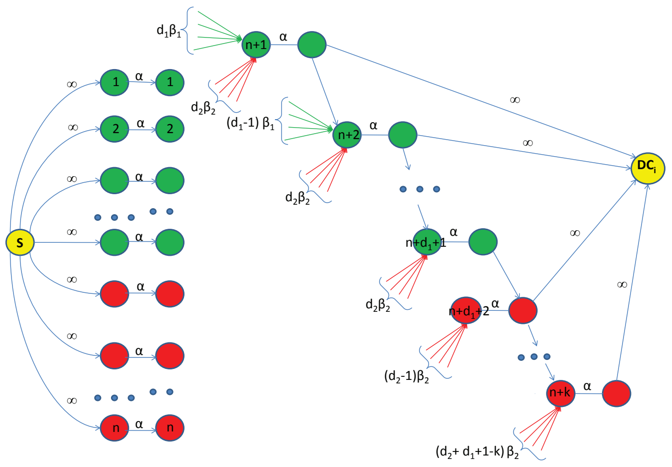

Fig. 3. $\mathcal{G}^*$ for $d_1 < k$

V. Scenario B: $d_1 < k$

In this case, the information flow graph $\mathcal{G}^*$ has a minimum min-cut similar to what is shown in Fig.3. As a result, according to min-cut theorem as is addressed in Section IV, the following condition should be satisfied,

$$\sum_{i=0}^{d_1} \min\{(d_1\beta_1 + d_2\beta_2 - i\beta_1),\ \alpha\} +$$
$$\textstyle\sum_{i=d_1+1}^{k-1} \min\{(d_1+d_2-i)\beta_2,\ \alpha\} \ \geq\ M \qquad (16)$$

The above condition introduces a tradeoff between α and β_2 which is computed as follows,

$$\alpha_{\text{min}}(d_1, d_2, k', \beta_2) =$$

$$\begin{cases} \frac{M}{k} & \beta_2 \in \big[f_1(0), \infty\big) \\ \frac{2M - g_1(i)\beta_2}{2(k-i)} & \beta_2 \in \big[f_1(i), f_1(i-1)\big) \\ \frac{2M - (g_1(k-d_1-1) + g_2(i))\beta_2}{2(d_1-i)} & \beta_2 \in \big[f_2(i), f_2(i-1)\big)\ , \end{cases}$$

where

$$\begin{aligned} f_1(i) &\triangleq \frac{2M}{2k(d-k) + (i+1)(2k-i))} \\ f_2(i) &\triangleq \frac{2M}{(2kd - k^2 - d_1^2 - d_1 + k + 2d_1k') + ik'(2d_1 - i - 1)} \\ g_1(i) &\triangleq i(2d - 2k + i + 1) \\ g_2(i) &\triangleq (i+1)(2d_2 + ik')\ . \end{aligned} \qquad (17)$$

Thus, $\beta_{2\text{min}}$ can be computed as,

$$\beta_{2_{min}} = f_2(d_1 - 1) = \frac{2M}{2kd - k^2 + k + (d_1^2 + d_1)(k' - 1)}\ . \qquad (18)$$

Accordingly, GMSR and GMBR, two extremal points of trade-off curve, have the following storage capacity-repair bandwidth,

$$(\alpha_{\text{GMSR}}, \gamma_{\text{GMSR}}) \ =\ (\frac{M}{k}, \frac{M(d_1k' + d_2)}{k(d-k+1)})\ . \qquad (19)$$

$$(\alpha_{\text{GMBR}}, \gamma_{\text{GMBR}}) \ =$$
$$(\tfrac{2M(d_1k'+d_2)}{2kd-k^2+k+(d_1^2+d_1)(k'-1)}, \tfrac{2M(d_1k'+d_2)}{2kd-k^2+k+(d_1^2+d_1)(k'-1)})\ . \qquad (20)$$

A. Comparison between GMSR and MSR when $d_1 < k$

Referring to (1) and (19), GMSR and MSR have an equal storage capacity per node. To get an insight regarding the repair bandwidth, we define the following repair bandwidth ratio,

$$\rho_{\text{MSR}}(k') \triangleq \frac{\gamma_{\text{GMSR}}(k')}{\gamma_{\text{MSR}}} = \frac{d_1k' + d_2}{d}\ . \qquad (21)$$

This ratio is always greater than one for $k' > 1$, meaning GMSR code imposes a large bandwidth to the system as compared to MSR. Similarly, the download cost ratio is defined as,

$$\eta_{\text{MSR}}(k') \triangleq \frac{C_{T_{\text{GMSR}}}(k')}{C_{T_{\text{MSR}}}} = \frac{C_1d_1k' + C_2d_2}{C_1d_1 + C_2d_2}\ . \qquad (22)$$

Again $\eta(k')$ for all positive values of k' is greater than one. Having larger repair bandwidth and storage capacity as well as higher download cost, one can conclude that GMSR does not have favorable result as compared to MSR. Thus, GMSR does not perform well for the case of $d_1 < k$, meaning in this case it is better to set $\beta_1 = \beta_2$ (MSR approach).

B. Comparison between GMBR and MBR when $d_1 < k$

As the storage per node is equal to the repair bandwidth for both MBR and GMBR codes, we concentrate on the repair bandwidth. Again, we define the repair bandwidth ratio $\rho_{\text{MBR}}(k')$ as follows:

$$\begin{aligned} \rho_{\text{MBR}}(k') &\triangleq \frac{\gamma_{\text{GMBR}}(k')}{\gamma_{\text{MBR}}} \\ &= \frac{(d_1k' + d_2)(2kd - k^2 + k)}{\big(2kd - k^2 + k + (d_1^2 + d_1)(k' - 1)\big)d}\ . \end{aligned} \qquad (23)$$

$\rho(k')$ has a positive derivative with respect to k' and noting $\rho(1) = 1$ it follows $\rho(k') \geq 1$ for $k' \geq 1$. Thus, MBR outperforms GMBR in terms of having lower repair band-width. Similarly, we define the download cost ratio as follows,

$$\begin{aligned} \eta_{\text{MBR}}(k') &\triangleq \frac{C_{T_{\text{GMBR}}}(k')}{C_{T_{\text{MBR}}}} \\ &= \frac{(C_1d_1k' + C_2d_2)(2kd - k^2 + k)}{(C_1d_1 + C_2d_2)\big(2kd - k^2 + k + (d_1^2 + d_1)(k' - 1)\big)}\ . \end{aligned} \qquad (24)$$

Again, to have the download cost of GMBR lower than that of MBR, the following condition should be satisfied,

$$\frac{C_2}{C_1} \geq \frac{2kd - k^2 + k - d_1^2 - d_1}{d_2(d_1 + 1)}\ . \qquad (25)$$

In this case, the minimum value of η is achieved as k' tends to infinity, i.e., $\eta(+\infty) = \frac{(C_1d_1)(2kd-k^2+k)}{(C_1d_1+C_2d_2)(d_1^2+d_1)}$

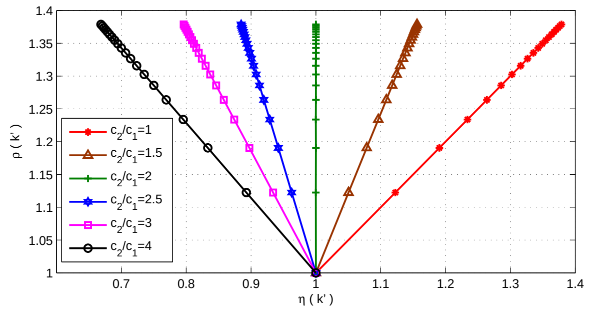

Fig. 4. The tradeoff curves between the relative cost and repair bandwidth ratio for GMSR code.

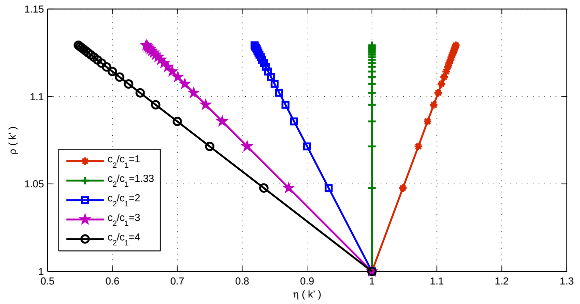

Fig. 5. The tradeoff curves between the relative cost and repair bandwidth ratio for GMBR code.

VI. Numerical Results

This section aims at providing some numerical results to get an insight regarding the proposed GMSR and GMBR codes and their advantages in terms of the corresponding storage capacity and/or repair bandwidth as compared to the MSR and MBR codes. In Fig.4, $\rho(k')$ versus $\eta(k')$ of the GMSR code for different integer values of k' in the interval $[1, 20]$ and for different relative cost ratios of $\frac{C_2}{C_1}$ is illustrated. Moreover, it is assumed $(n, k, d_1, d_2) = (15, 5, 8, 6)$, which corresponds to scenario A, since $d_1 \geq k$. Noting the condition (12), in this example, if $\frac{C_2}{C_1} \geq \frac{d_1}{d_1-k+1} = 2$, the download cost of GMSR is lower than that of MSR ($\eta(k') \leq 1$). This is in accordance to what is inferred from Fig.4. Moreover, Fig.4 depicts the amount of increment in repair bandwidth for a given download cost ratio. Similarly, Fig.5 provides the same result for GMBR with the same parameters, i.e., $(n, k, d_1, d_2) = (15, 5, 8, 6)$. Again, referring to equation (15), $\eta(k') \geq 1$ for $\frac{C_2}{C_1} \geq \frac{2d_1}{2d_1-k+1} = 1.33$ which is in accordance to the result of Fig.5.

Fig.6 depicts the $\rho(k')$ versus $\eta(k')$ for GMBR when $(n, k, d_1, d_2) = (15, 5, 4, 10)$. Noting $d_1 < k$, this case belongs to scenario B. Referring to (25), if $\frac{C_2}{C_1} > \frac{2kd-k^2+k-d_1^2-d_1}{d_2(d_1+1)} = 2$, the downlod cost of GMBR is lower than that of MBR ($\eta(k') \leq 1$). Fig.6 confirms this threshold for $\frac{C_2}{C_1}$. Moreover, it shows how download cost ratio affects the repair bandwidth ratio $\left(\rho(k')\right)$.

Also, the tradeoff curves between the storage capacity per node and repair bandwidth for RC and GRC codes for two different values of $k' = 2, 4$ are shown in Fig.7. This shows the storage capacity-repair bandwidth tradeoff curve of RC code outperforms that of GRC (the dotted curve), while as is noted before, GRC may result in lower download cost as compared to that of RC code.

Finally, Fig.8 is provided to show the impact of different values of k' on η and for different values of $\frac{C_2}{C_1}$. Fig.8 confirms that under certain conditions as is mentioned in the preceding sections, $\eta(k')$ is a decreasing function with respect to k'.

VII. Conclusion

This paper aims at addressing the cost bandwidth tradeoff in distributed storage systems when the download cost of storage nodes are not the same. Specifically, we concentrate

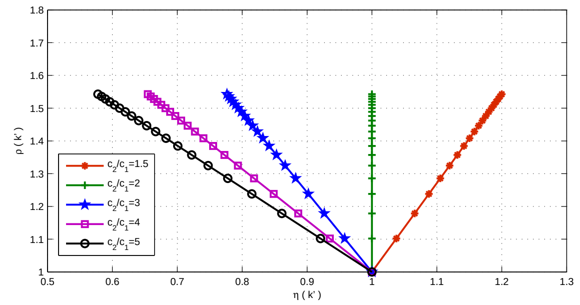

Fig. 6. The tradeoff curves between the relative cost and bandwidth ratio for GMBR code.

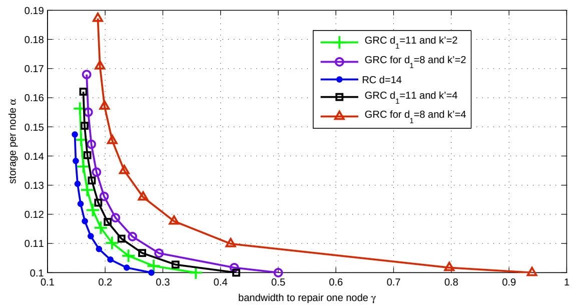

Fig. 7. The tradeoff curves between the storage per node and repair bandwidth.

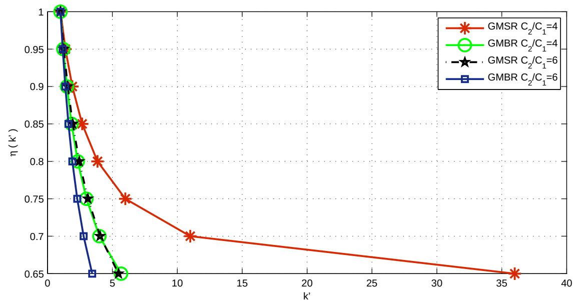

Fig. 8. The effect of k' on the relative cost.

to case that there are two sets of nodes, each having different download costs. Accordingly, using the corresponding *Information Flow Graph*, a new variation of regenerating codes, called generalized regenerating codes, is proposed and is shown under some certain conditions outperform the current regenerating codes in terms of having lower download cost, while having a marginal increase in the repair bandwidth.

REFERENCES

[1] A. G. Dimakis, P. G. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *Submitted to IEEE Transactions of Information Theory*.

[2] R. Bhagwan, K. Tati, Y. C. Cheng, S. Savage, and G. M. Voelker, "Total recall:system support for automated availability management," *NSDI*, 2004.

[3] F. Dabek, J. Li, E. Sit, J. Robertson, M. Kaashoek, and R. Morris, "Designing a dht for low latency and high throughput," 2004.

[4] R. Rodriguez and B. Liskov, "High availability in dhts: Erasure coding vs. replication," *in Proc. IPTPS*, 2005.

[5] H. Weatherspoon and J. D. Kubiatowicz, "Erasure coding vs. replication:a quantitative comparison," *in Proc. IPTPS*, 2002.

[6] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions of Information Theory*, vol. 46, pp. 1204–1216, July 2000.

[7] S. Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions of Information Theory*, vol. 49, pp. 371–381, February 2003.

[8] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, October 2003.

VIII. APPENDIX

To derive the optimal tradeoff between α and β_2, one can fix β_2 and d_1, d_2, k' (to some integer values) and then find the minimum value of α such that (4) and (16) are satisfied. To this end, we define α_{min} as follows,

$$\begin{aligned} \alpha_{\text{min}} \quad & (d_1, d_2, k', \beta_2) \triangleq \min \alpha \\ \text{subject to} \quad & : C \ \geq \ M \ , \end{aligned} \tag{26}$$

where depending to on the condition that $d_1 \geq k$ or $d_1 < k$ we have,

$$\begin{aligned} C \ &\triangleq \ \sum_{i=0}^{k-1} \min\{(d_1\beta_1 + d_2\beta_2 - i\beta_1),\ \alpha\} \text{ for } d_1 \geq k \\ C \ &\triangleq \ \sum_{i=0}^{d_1} \min\{(d_1k' + d_2 - ik')\beta_2,\ \alpha\} \\ &+ \ \sum_{i=d_1+1}^{k-1} \min\{(d_1 + d_2 - i)\beta_2,\ \alpha\} \text{ for } d_1 < k \end{aligned} \tag{27}$$

The result of $d_1 \geq k$:
To prove (5), substituting $\beta_1 = k'\beta_2$ in the corresponding C (equation (27) with $d_1 \geq k$), it follows,

$$\begin{aligned} C &\triangleq \sum_{i=0}^{k-1} \min\{(d_1\beta_1 + d_2\beta_2 - i\beta_1),\ \alpha\} \\ &= \sum_{i=0}^{k-1} \min\{(d_1k' + d_2 - ik')\beta_2,\ \alpha\} \ \geq \ M \ . \end{aligned} \tag{28}$$

Thus, C can be computed, assuming α belongs to one of the following intervals,

$$C(\alpha) =$$

$$\begin{cases} k\alpha & \alpha \in \big[0, h(1)\beta_2\big] \\ (k-1)\alpha + h(1)\beta_2 & \alpha \in \big(h(1)\beta_2, h(2)\beta_2\big] \\ \vdots & \\ (k-j)\alpha + \sum_{i=1}^{j} h(i)\beta_2 & \alpha \in \big(h(j)\beta_2, h(j+1)\beta_2\big] \\ \vdots & \\ \alpha + \sum_{i=1}^{k-1} h(i)\beta_2 & \alpha \in \big(h(k-1)\beta_2, h(k)\beta_2\big] \ , \end{cases}$$

where

$$h(i) \triangleq d_1k' + d_2 - (k-i)k' \tag{29}$$

As a result, noting $C \geq M$, it follows,

$$\alpha_{\text{min}} =$$

$$\begin{cases} \frac{M}{k} & M \in \big[0, kh(1)\beta_2\big] \\ \frac{M - \left(\sum_{i=1}^{j} h(i)\right)\beta_2}{(k-j)} & M \in \big((k-j)h(j)\beta_2 + \sum_{i=1}^{j} h(i)\beta_2, \\ & \quad (k-j)h(j+1)\beta_2 + \sum_{i=1}^{j} h(i)\beta_2\big] \ , \\ & \quad j = 0, 1, ..., k-1 \end{cases}$$

or equivalently,

$$\alpha_{\text{min}} =$$

$$\begin{cases} \frac{M}{k} & \beta_2 \in \big[\frac{M}{kh(1)}, \infty\big) \\ \frac{M - \left(\sum_{i=1}^{j} h(i)\right)\beta_2}{(k-j)} & \beta_2 \in \big[\frac{M}{(k-j)h(j+1) + \sum_{i=1}^{j} h(i)}, \\ & \quad \frac{M}{(k-j)h(j) + \sum_{i=1}^{j} h(i)}\big),\ j = 0, 1, ..., k-1 \end{cases}$$

As a result, noting (29), it follows,

$$\alpha_{\text{min}} =$$

$$\begin{cases} \frac{M}{k} & \beta_2 \in \big[f(0), \infty\big) \\ \frac{2M - g(i)\beta_2}{2(k-i)} & \beta_2 \in \big[f(i), f(i-1)\big) \ , \qquad i = 0, 1, ..., k-1, \end{cases}$$

where

$$f(i) \triangleq \frac{2M}{2kh(0) + (i+1)(2k-i)k'} \tag{30}$$

$$g(i) \triangleq i(2d_1k' + 2d_2 - 2kk' + (i+1)k') \tag{31}$$

$$\beta_{\text{min}} = f(k-1) \tag{32}$$

The result of $d_1 < k$:

In this case, substituting $\beta_1 = k'\beta_2$ in C (Equation (27) when $d_1 < k$) and following the same approach as the case of $d_1 \geq k$, it follows,

$$C(\alpha) =$$

$$\begin{cases} k\alpha & \alpha \in [0, h(1,0)\beta_2] \\ (k-1)\alpha + h(1,0)\beta_2 & \alpha \in (h(1,0)\beta_2, h(2,0)\beta_2] \\ \vdots & \\ (k-j)\alpha + \sum_{i=1}^{j} h(i,0)\beta_2 & \alpha \in (h(j,0)\beta_2, h(j+1,0)\beta_2] \\ \vdots & \\ d_1\alpha + \sum_{i=1}^{k-d_1} h(i,0)\beta_2 & \alpha \in (h(k-d_1,0)\beta_2, \\ & \quad h(k-d_1,1)\beta_2] \\ A & \alpha \in (h(k-d_1,1)\beta_2, \\ & \quad h(k-d_1,2)\beta_2] \\ \vdots & \\ B & \alpha \in (h(k-d_1,t)\beta_2, \\ & \quad h(k-d_1,t+1)\beta_2] \\ \vdots & \\ D & \alpha \in (h(k-d_1,d_1-1)\beta_2, \\ & \quad h(k-d_1,d_1)\beta_2] \end{cases}$$

where

$$\begin{aligned} h(x,y) &\triangleq d_1 + d_2 - k + x + yk' \\ A &= (d_1-1)\alpha + \sum_{i=1}^{k-d_1} h(i,0)\beta_2 + h(k-d_1,1)\beta_2 \\ B &= (d_1-t)\alpha + \sum_{i=1}^{k-d_1} h(i,0)\beta_2 + \sum_{i=1}^{t} h(k-d_1,i)\beta_2 \\ D &= \alpha + \sum_{i=1}^{k-d_1} h(i,0)\beta_2 + \sum_{i=1}^{d_1-1} h(k-d_1,i)\beta_2 \\ C(\alpha_{\text{min}}) &= M \ . \end{aligned} \tag{33}$$

Thus, $\alpha_{\text{min}} = C^{-1}(M)$ can be computed as,

$$\alpha_{\text{min}}(d_1, d_2, k', \beta_2) =$$

$$\begin{cases} \frac{M}{k} & \beta_2 \in \big[f_1(0), \infty\big) \\ \frac{2M - g_1(i)\beta_2}{2(k-i)} & \beta_2 \in \big[f_1(i), f_1(i-1)\big) \\ \frac{2M - (g_1(k-d_1-1) + g_2(i))\beta_2}{2(d_1-i)} & \beta_2 \in \big[f_2(i), f_2(i-1)\big) \ , \end{cases}$$

where

$$\begin{aligned} f_1(i) &\triangleq \frac{2M}{2k(d-k) + (i+1)(2k-i))} \\ f_2(i) &\triangleq \frac{2M}{(2kd - k^2 - d_1^2 - d_1 + k + 2d_1k') + ik'(2d_1 - i - 1)} \\ g_1(i) &\triangleq i(2d - 2k + i + 1) \\ g_2(i) &\triangleq (i+1)(2d_2 + ik') \ . \end{aligned} \tag{34}$$

Finally, $\beta_{2\text{min}}$ can be computed as,

$$\beta_{2_{min}} = f_2(d_1 - 1) \tag{35}$$